Chapter 5 part 2
LINK LAYER

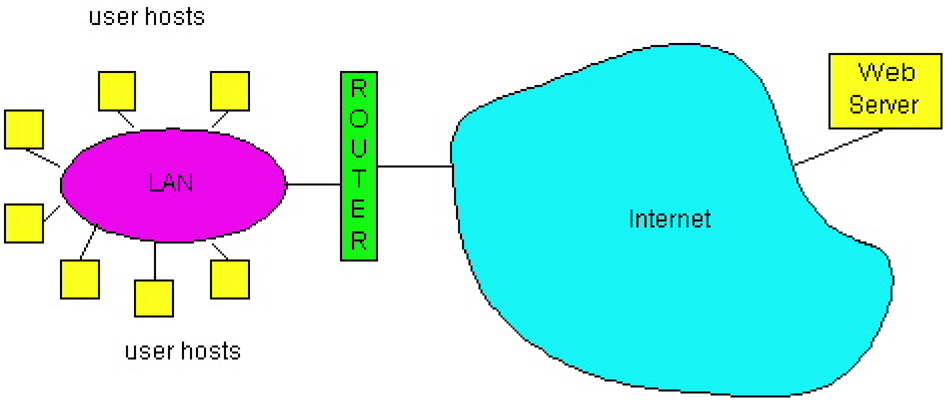
Computer Networks

Timothy Roscoe

Summer 2007

LAN technologies

- Data link layer so far
  - services, error detection/correction, multiple access
- Next: LAN technologies
  - LAN addressing, ARP
  - Ethernet
  - hubs, bridges, switches
  - PPP



LAN Addresses and ARP

32-bit IP address

- *network-layer* address
- used to get datagram to destination network (recall IP network definition)

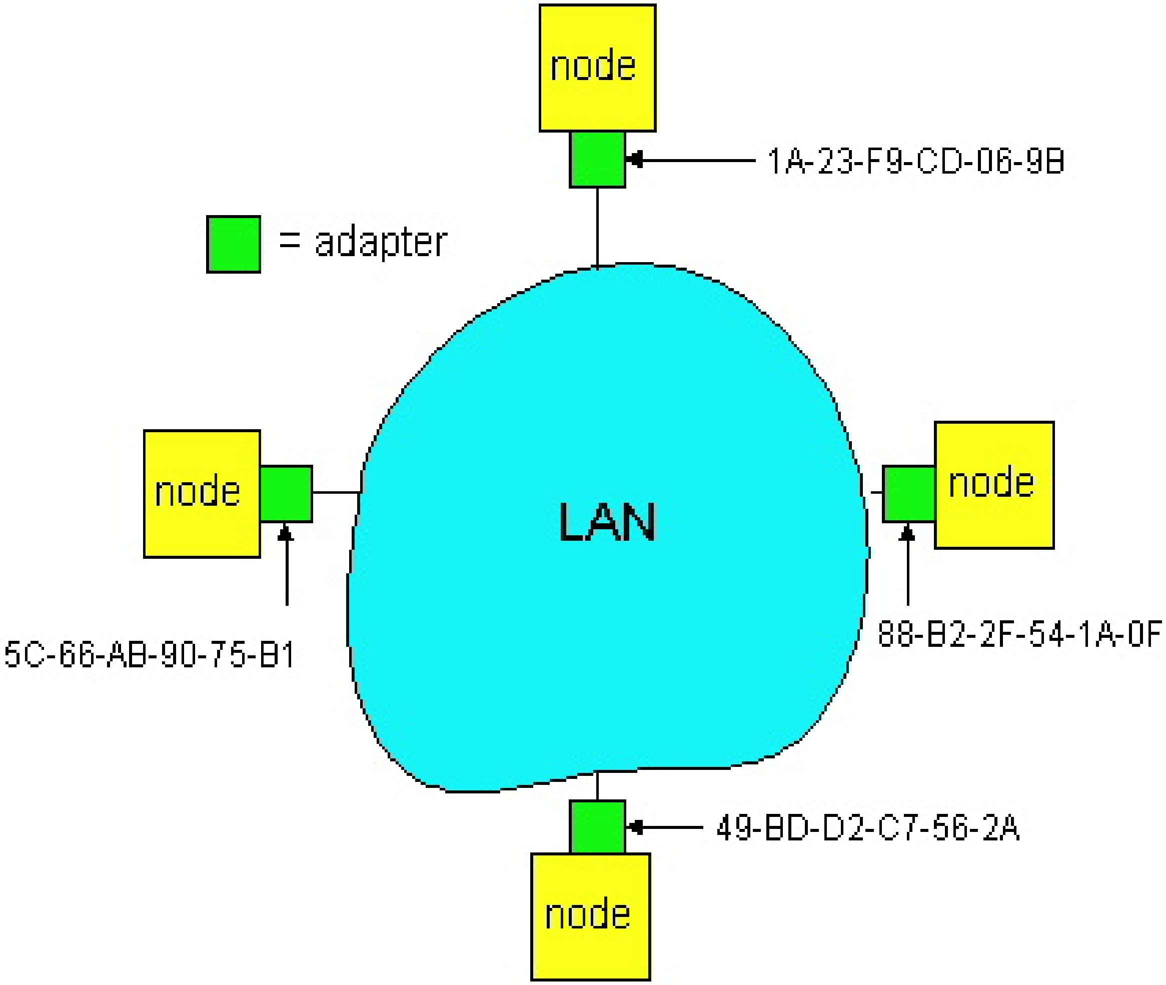
MAC (or LAN or physical) address

- used to get datagram from one interface to another physically-connected interface (same LAN)
- 48 bit MAC address (for most LANs) burned in the adapter ROM

ARP (Address Resolution Protocol)

- IP-Address → MAC-Address

LAN Addresses and ARP



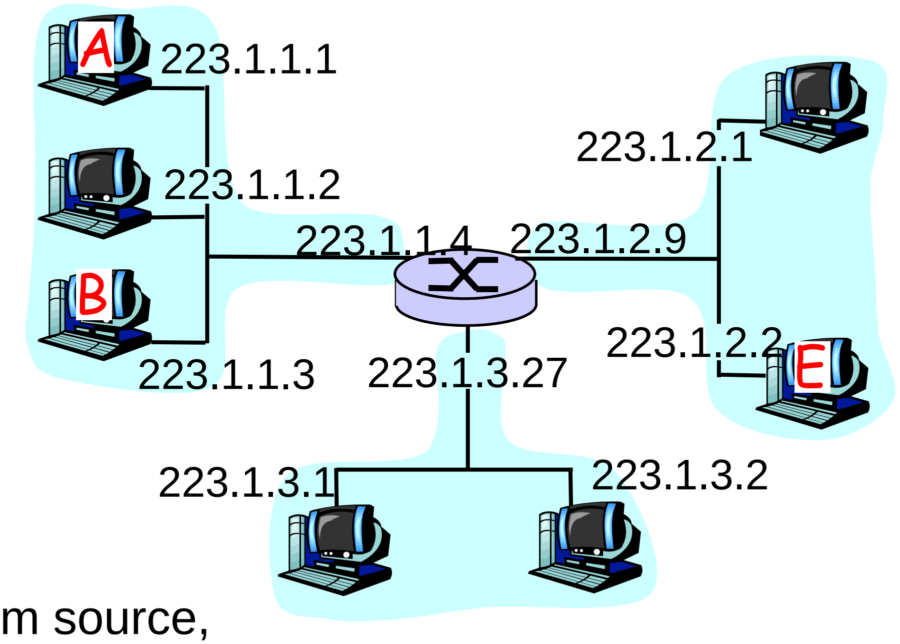
Each adapter on LAN has unique LAN address

LAN Address (more)

- MAC address allocation administered by IEEE
- manufacturer buys portion of MAC address space (to assure uniqueness)
- Analogy
  - MAC address like Social Security (AHV) Number
  - IP address like postal address
- MAC flat address → portability
  - can move LAN card from one LAN to another
- IP hierarchical address NOT portable
  - depends on network to which one attaches

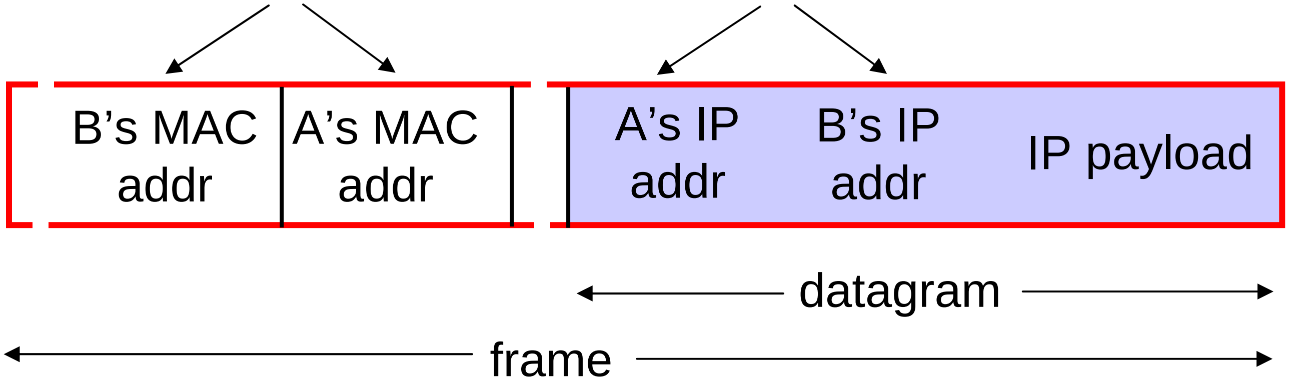
Recall earlier routing discussion

- Starting at A
- given IP datagram addressed to B
- look up network address of B
- find B on same net as A
- link layer send datagram to B inside link-layer frame



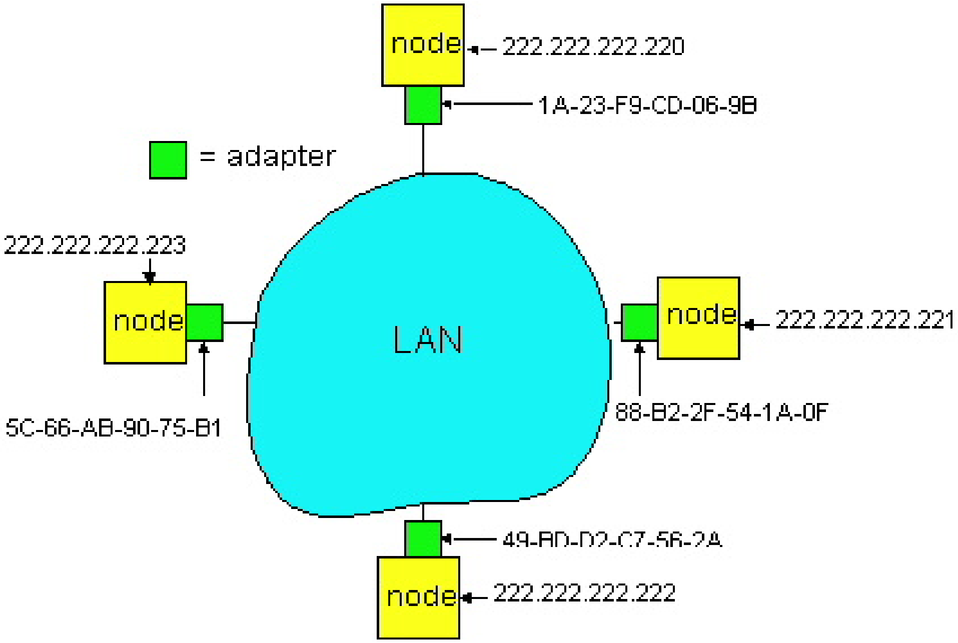
frame source, dest address

datagram source, dest address



ARP: Address Resolution Protocol

Question: How to determine MAC address of B given B’s IP address?



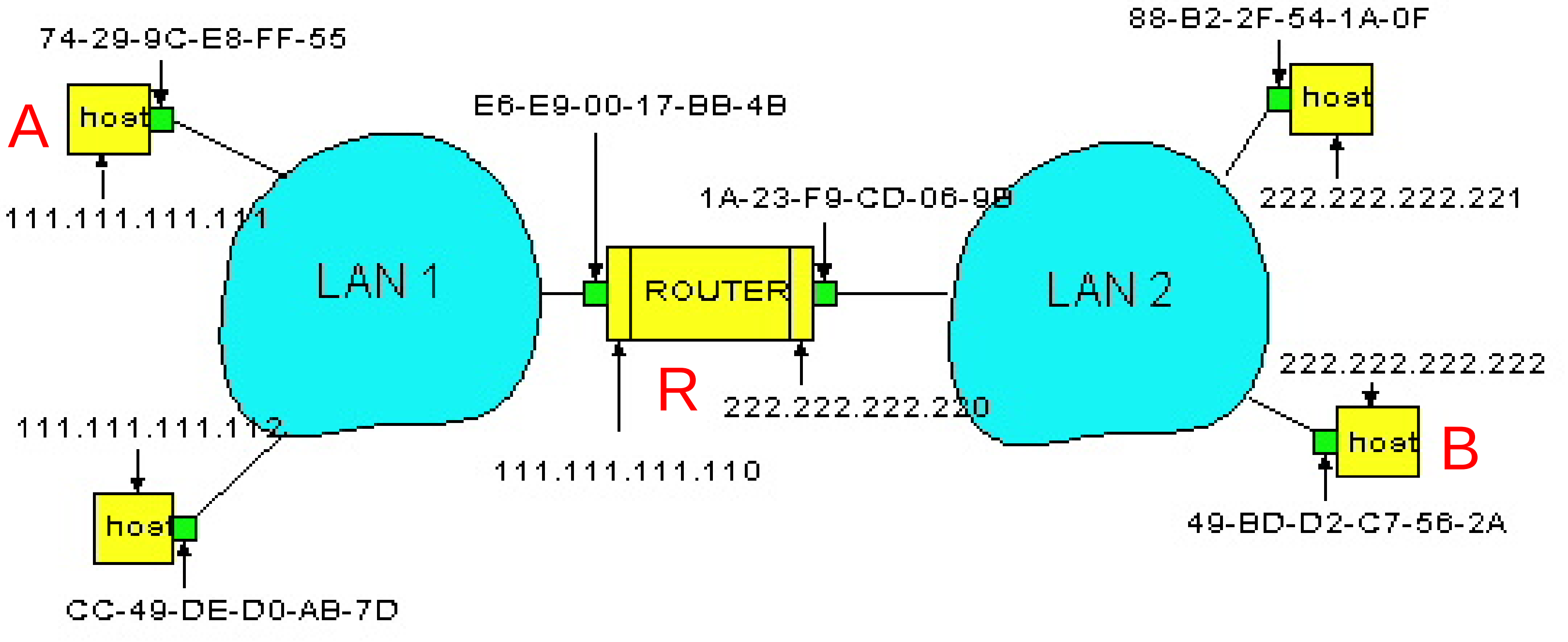
- Each IP node (Host, Router) on LAN has ARP table
- ARP Table: IP/MAC address mappings for some LAN nodes
  < IP addr; MAC addr; TTL >
- TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)

ARP protocol

- A knows B's IP address, wants to learn physical address of B
- A broadcasts ARP query packet, containing B's IP address
  - all machines on LAN receive ARP query
- B receives ARP packet, replies to A with its (B's) physical
  layer address
- A caches (saves) IP-to-physical address pairs until information becomes old (times out)
- This is a so-called “soft state” protocol
  - information times out (goes away) unless refreshed

Routing to another LAN

walkthrough: routing from A to B via R



- In routing table at source host, find router 111.111.111.110
- In ARP table at source, find MAC address E6-E9-00-17-BB-4B, etc.

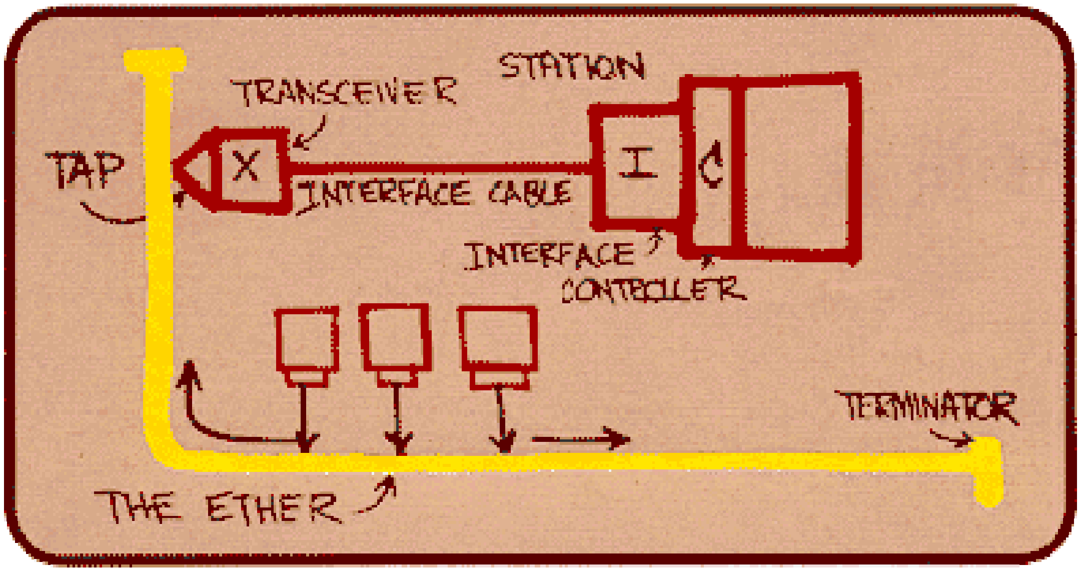
Continued…

- A creates IP packet with source A, destination B
- A uses ARP to get R’s physical layer address for 111.111.111.110
- A creates Ethernet frame with R's physical address as destination, Ethernet frame contains A-to-B IP datagram
- A’s data link layer sends Ethernet frame
- R’s data link layer receives Ethernet frame
- R removes IP datagram from Ethernet frame,
  sees its destined to B
- R uses ARP to get B’s physical layer address
- R creates frame containing A-to-B IP datagram sends to B

Ethernet

Currently predominant LAN technology

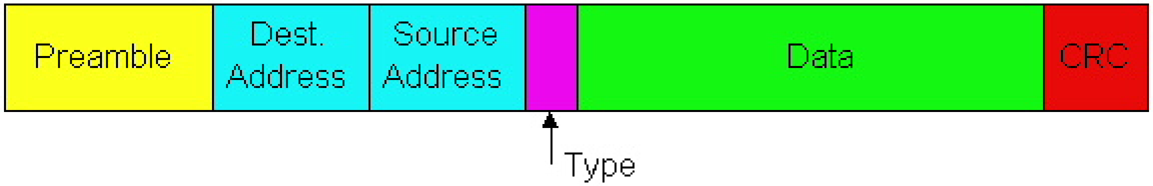
- cheap: $2 for 100Mbps!
- first widely used LAN technology
- Simpler/cheaper than token rings and ATM
- Keeps up with speed race: 10, 100, 1000, 10000 Mbps



Metcalfe's Ethernet sketch

Ethernet Frame Structure

- Sending adapter encapsulates IP datagram (or other network layer protocol packet) in *Ethernet frame*



- Preamble
  - 7 bytes with pattern 10101010
  - Followed by 1 byte with pattern 10101011
  - Used to synchronize receiver, sender clock rates
- Addresses
  - 6 bytes, frame is received by all adapters on a LAN and dropped if address does not match
- Type (2 bytes): indicates the higher layer protocol, mostly IP (0x0800) but others may be supported such as Novell IPX and AppleTalk)
- CRC (4 bytes): checked at receiver, if error is detected, the frame is simply dropped

CSMA/CD (connectionless & unreliable)

- Connectionless
  - No handshaking between sending and receiving adapter.
- Unreliable
  - receiving adapter doesn’t send ACKs or NAKs to sending adapter
  - stream of datagrams passed to network layer can have gaps, which will be filled if app is using TCP or seen by application
- No slots
- Carrier sense: adapter doesn’t transmit if it senses that some other adapter is transmitting
- Collision detection: transmitting adapter aborts when it senses that another adapter is transmitting
- Random access: Before attempting a retransmission, adapter waits a random time

Ethernet CSMA/CD algorithm

1. Adapter gets datagram from network layer and creates frame

2. If adapter senses channel idle, it starts to transmit frame. If it senses channel busy, waits until channel idle and then transmits

3. If adapter transmits entire frame without detecting another transmission, the adapter is done with frame !

4. If adapter detects another transmission while transmitting, aborts and sends jam signal

5. After aborting, adapter enters exponential backoff: after the m^{th} collision, adapter chooses a K at random from $\{0,1,2,\ldots,2^{m}-1\}$. Adapter waits K¢512 bit times and returns to Step 2

Ethernet's CSMA/CD (more)

Jam Signal

- make sure all other transmitters are aware of collision; 48 bits;

Bit time

- 0.1 microsec for 10 Mbps Ethernet
- for K=1023, wait time is about 50 msec

Exponential Backoff

- Goal: adapt retransmission attempts to estimated current load
  - heavy load: random wait will be longer
- first collision: choose K from {0,1}; delay is K¢512 bit transmission times
- after second collision: choose K from {0,1,2,3}
- after ten collisions, choose K from {0,1,2,3,4,…,1023}

CSMA/CD efficiency

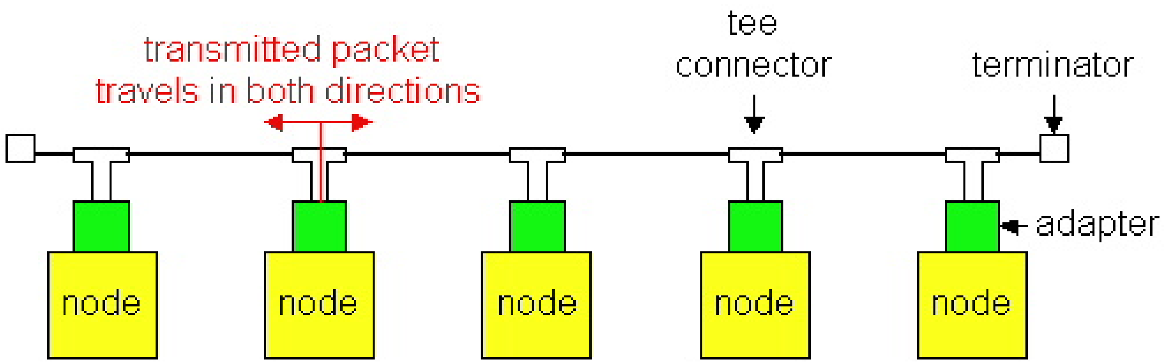
- t_{prop} = max. propagation time between any two nodes in LAN
- t_{trans} = time to transmit max-size frame

$$\text{utilization} \approx \frac{1}{1 + 6.2 \cdot t_{prop} / t_{trans}}$$

- Derivation of this formula is not trivial (not in this course)
- Remarks
  - Utilization goes to 1 as t_{prop} goes to 0
  - Utilization goes to 1 as t_{trans} goes to infinity
  - Much better than ALOHA, but still decentralized, simple, and cheap

Ethernet Technologies: 10Base2

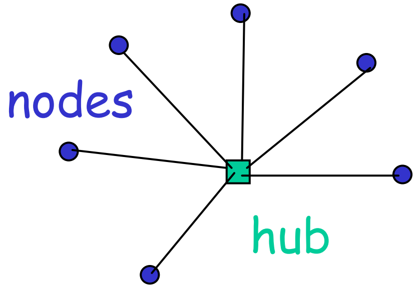
- 10: 10Mbps; 2: under 200 meters maximal cable length
- thin coaxial cable in a bus topology



- repeaters used to connect up to multiple segments
- repeater repeats bits it hears on one interface to its other interfaces: physical layer device only!
- has become a legacy technology

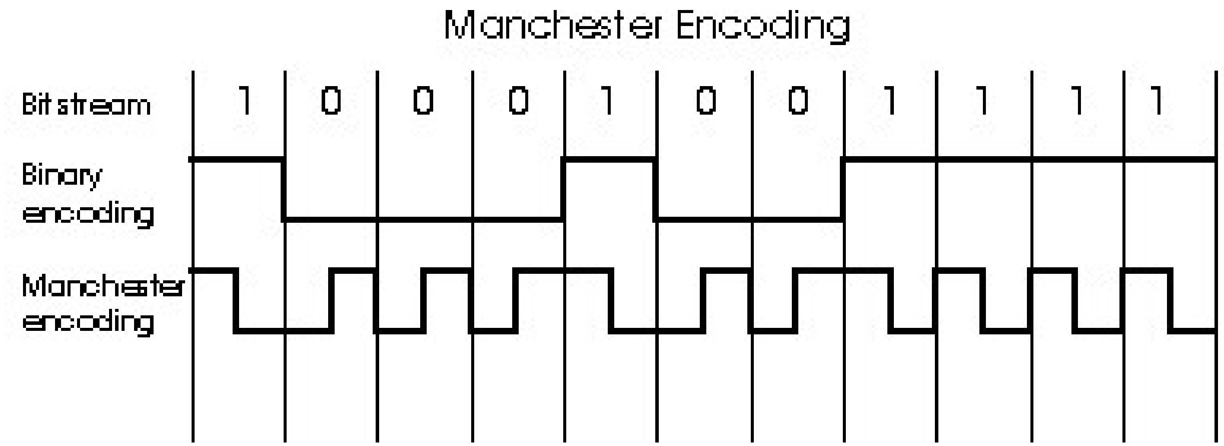
10BaseT and 100BaseT

- 10/100 Mbps rate; latter a.k.a. “fast ethernet”
- T stands for Twisted Pair
- Nodes connect to a hub: “star topology”; 100 m max distance between nodes and hub



- Hubs are essentially physical-layer repeaters
  - bits coming in on one link go out on all other links
  - no frame buffering
  - no CSMA/CD at hub: adapters detect collisions
  - provides net management functionality

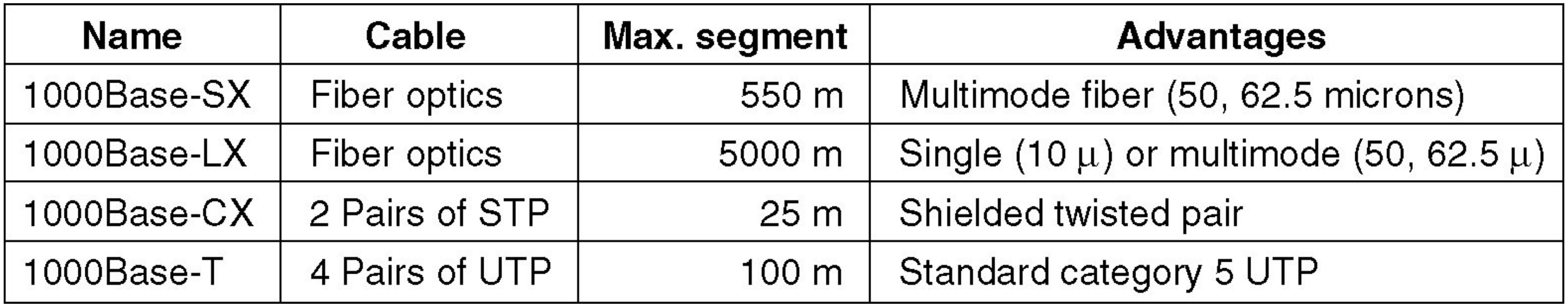
Manchester encoding



- Used in 10BaseT, 10Base2
- Each bit has a transition
- Allows clocks in sending and receiving nodes to synchronize to each other
    - – no need for a centralized, global clock among nodes!
- Hey, this is physical-layer stuff!

Gbit Ethernet

- Standard Ethernet frame format
  - plus “Jumbo frames”
- point-to-point links and shared broadcast channels
  - CSMA/CD is used for shared mode
  - short distances between nodes to be efficient
  - Full-Duplex at 1 Gbps for point-to-point links



Name	Cable	Max. segment	Advantages
1000Base-SX	Fiber optics	550 m	Multimode fiber (50, 62.5 microns)
1000Base-LX	Fiber optics	5000 m	Single (10 μ) or multimode (50, 62.5 μ)
1000Base-CX	2 Pairs of STP	25 m	Shielded twisted pair
1000Base-T	4 Pairs of UTP	100 m	Standard category 5 UTP

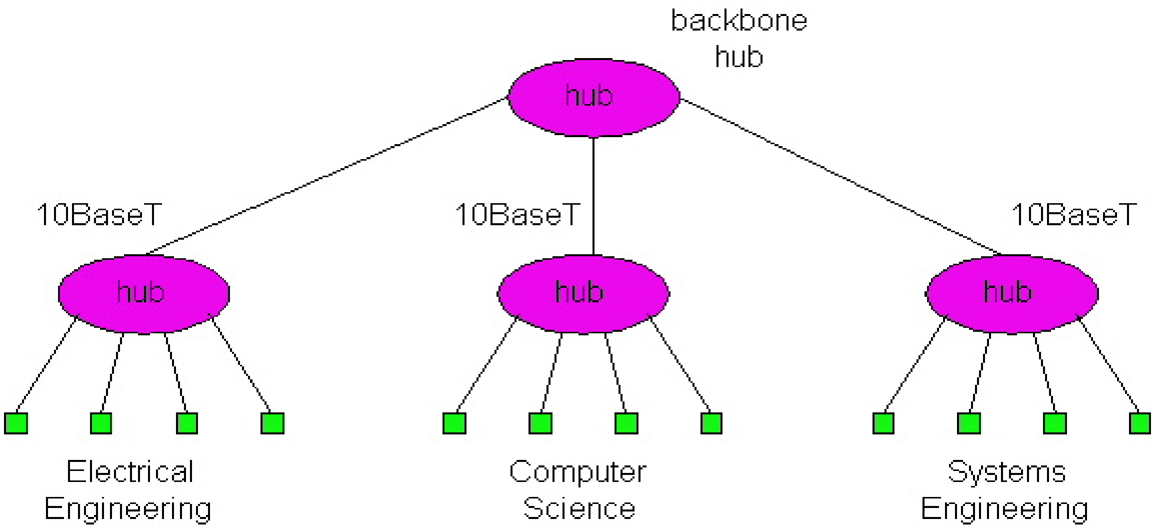
- 10 Gig ethernet now available…

Interconnecting LANs: overview

- Low-level: Hubs
- True link-layer: Bridges, Switches
  - Not routers or IP-layer connectivity (here)
- Addressing
- Forwarding
- “Routing” (topology management)
- Link-layer adaptation
  - E.g. Ethernet <-> 802.11

Interconnecting LANs: Hubs

- Backbone hub interconnects LAN segments
- Extends max. distance between nodes
- But individual segment collision domains become one large collision domain
  - if a node in CS and a node in EE transmit at same time: collision

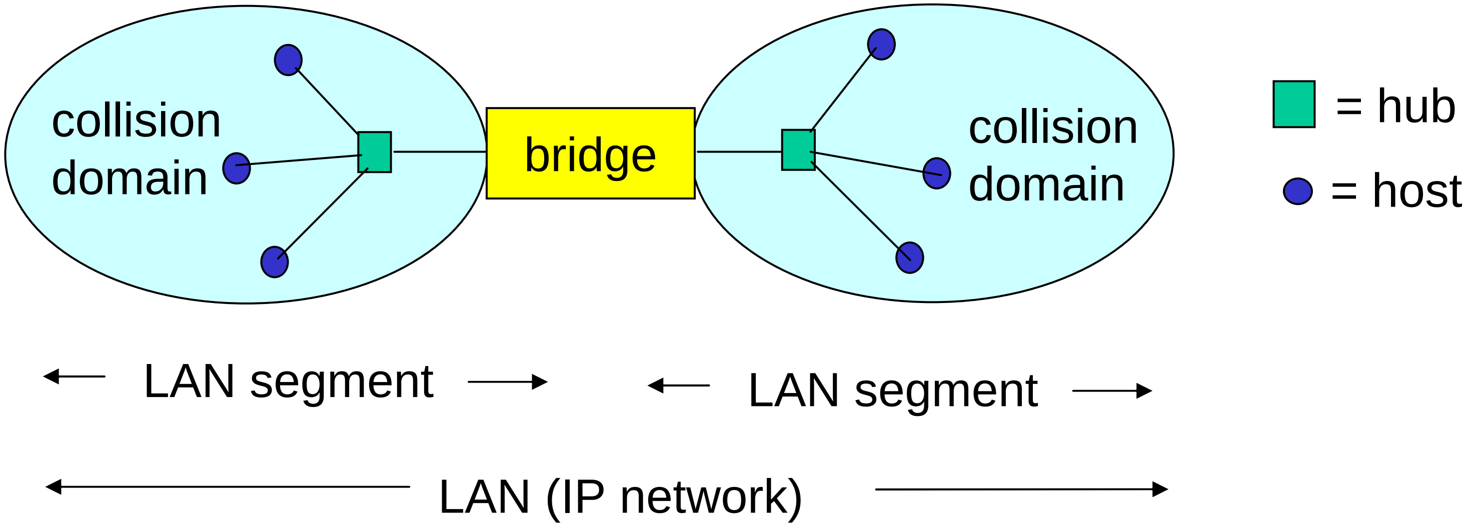


Interconnecting with Bridges

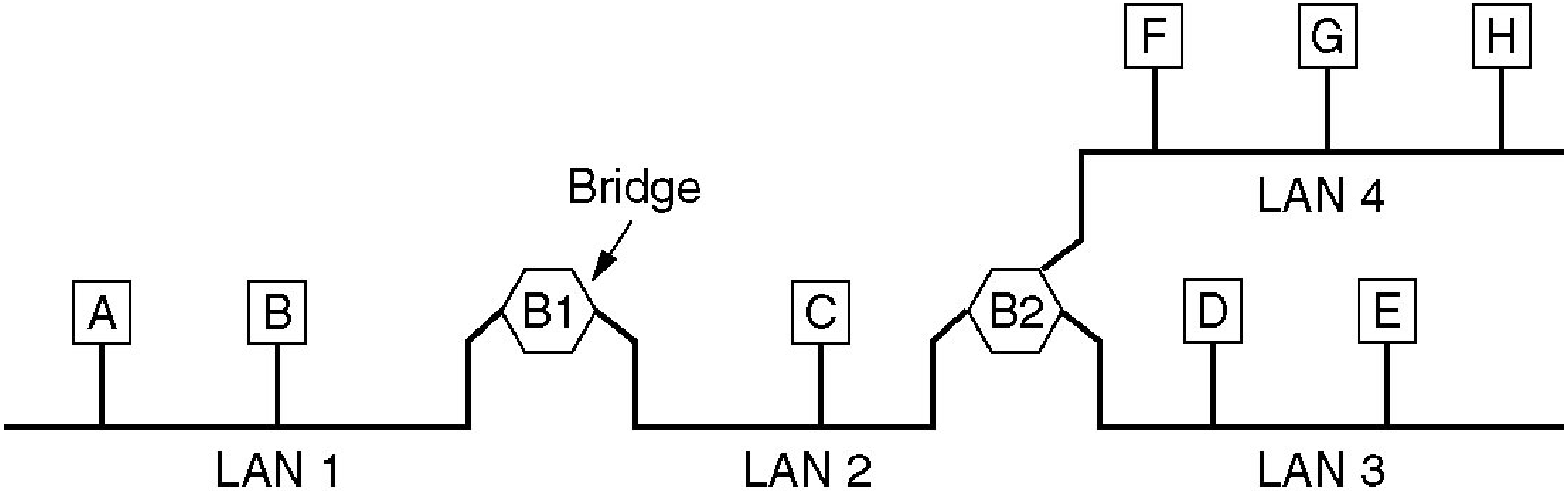
- A bridge is a link layer device
  - stores and forwards Ethernet frames
  - examines frame header and *selectively* forwards frame based on MAC destination address
  - when frame is to be forwarded on segment, uses CSMA/CD to access segment
- transparent
  - hosts are unaware of presence of bridges
- plug-and-play, self-learning
  - bridges do not need to be configured
- Switch: just a bridge with many ports (almost)

Bridges: traffic isolation

- Bridge installation breaks LAN into LAN segments
- Bridges *filter* packets:
  - same-LAN-segment frames not usually forwarded onto other LAN segments
  - segments become separate *collision domains*



Forwarding



How do determine to which LAN segment to forward frame?

Looks like a routing problem...

Self learning

- A bridge has a *bridge table*
  - with entries (Node LAN Address, Bridge Interface, Time Stamp)
  - stale entries in table dropped (TTL can be 60 min)

- bridges *learn* which hosts can be reached through which interfaces
  - when frame received, bridge “learns” location of sender, incoming LAN segment
  - records sender/location pair in bridge table

Filtering/Forwarding

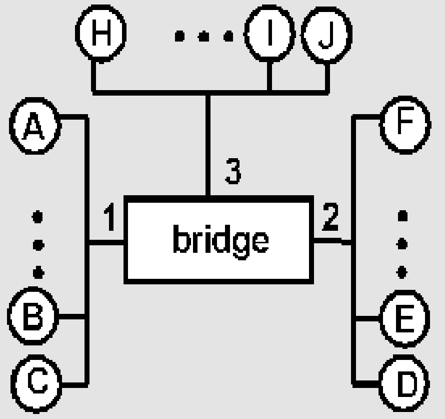
When bridge receives a frame:

```
index bridge table using MAC destination address
if entry found for destination
  then
    if dest on segment from which frame arrived
      then drop the frame
        else forward the frame on interface indicated
  else
    forward on all but the interface on which frame arrived
```

Bridge example

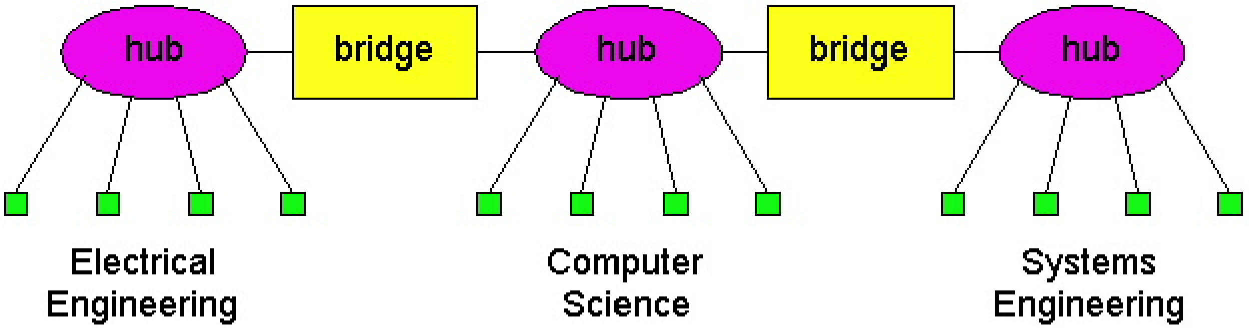
Suppose C sends frame to D and D replies back with frame to C.

- Bridge receives frame from C
  - notes in bridge table that C is on interface 1
  - because D is not in table, bridge sends frame into interfaces 2 and 3
- frame received by D
- D generates frame for C, sends
- bridge receives frame
  - notes in bridge table that D is on interface 2
  - bridge knows C is on interface 1, so *selectively* forwards frame to interface 1



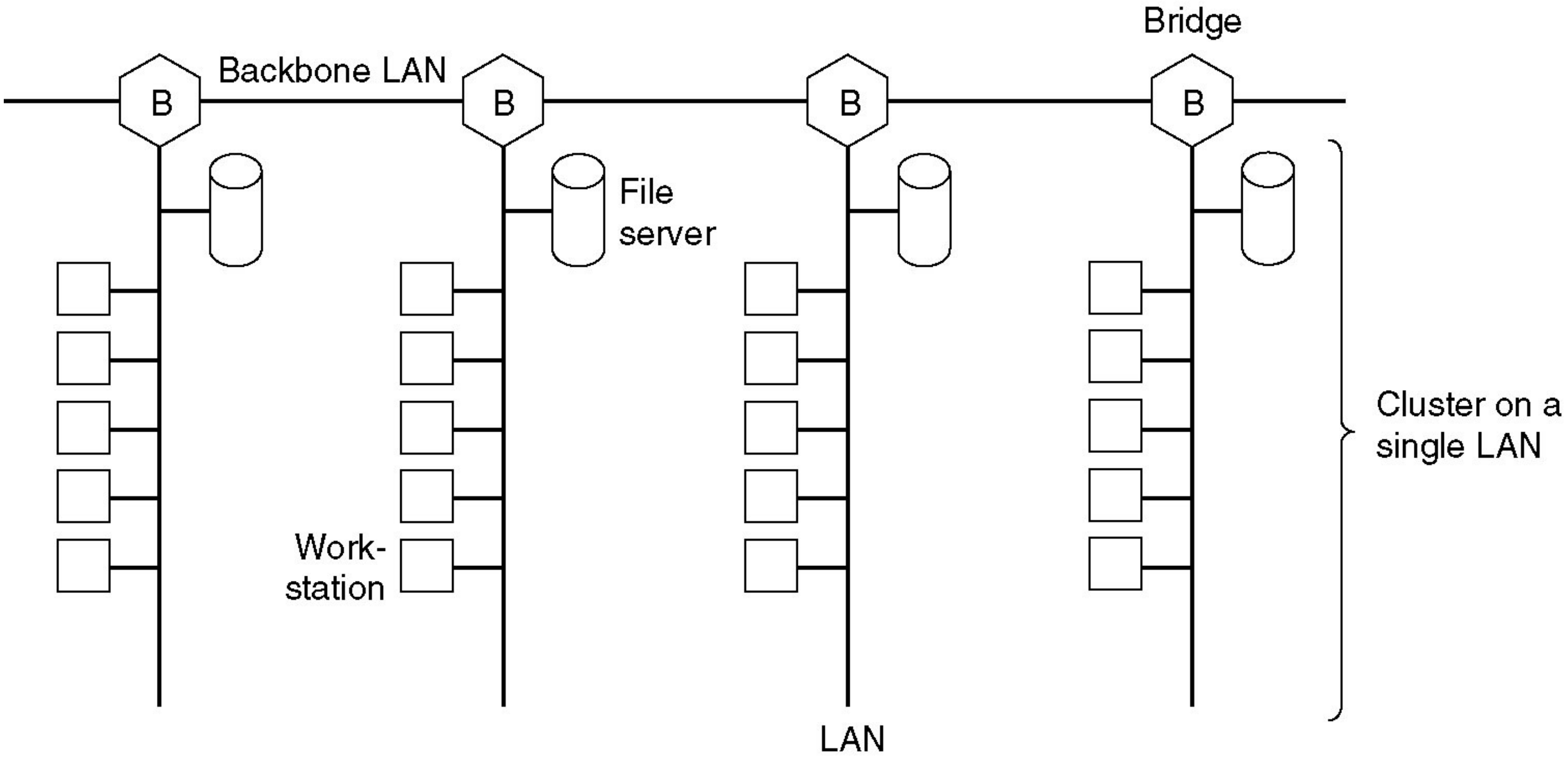
address	port
A	1
B	1
E	2
H	3
J	3

Interconnection without backbone



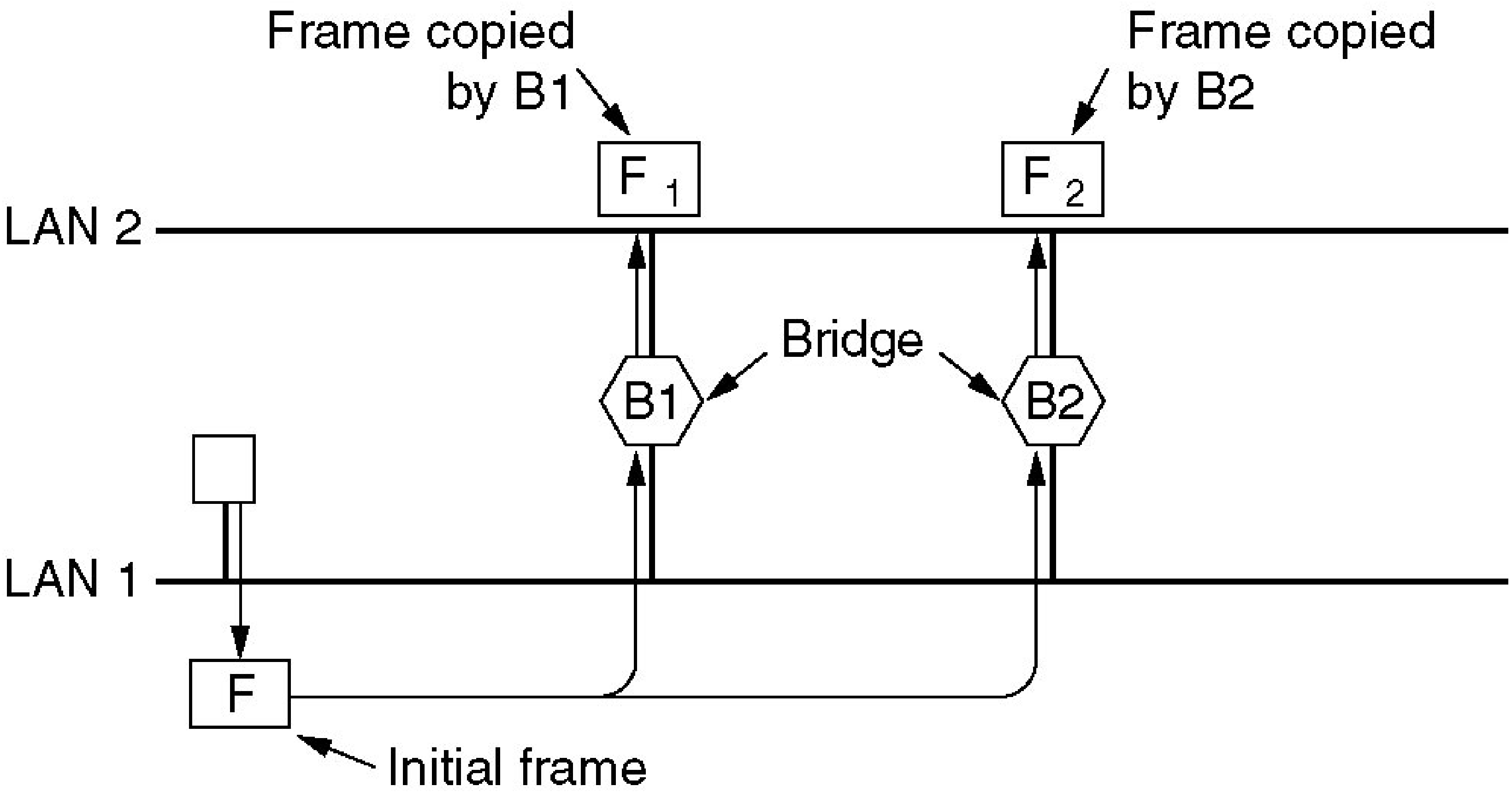
- Not recommended for two reasons:
  - single point of failure at Computer Science hub
  - all traffic between EE and SE must pass CS segment

Backbone configuration

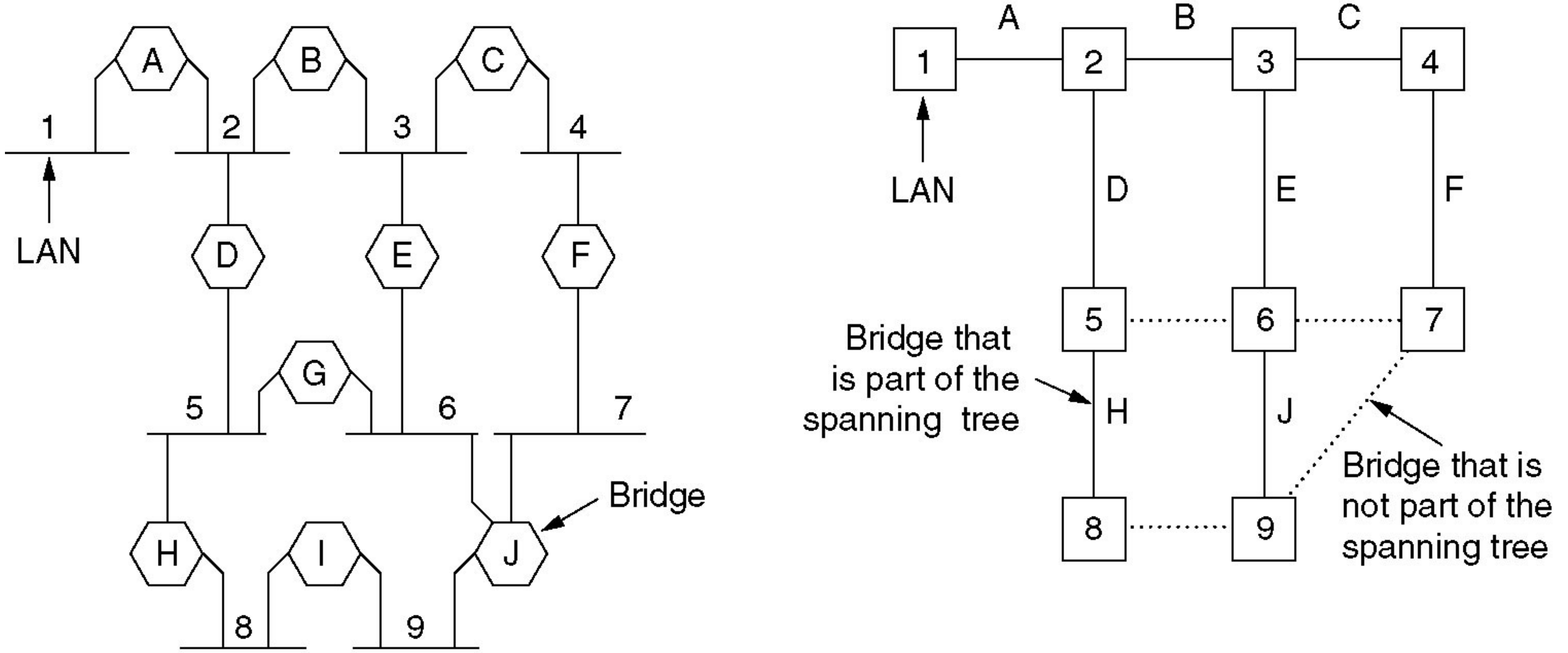


Recommended!

Problem with bridges: Loops!



Bridges: Spanning Tree


A
B
C
1
2
3
4
LAN
D
E
F
G
5
6
7
Bridge
H
I
J
8
9
A
B
C
1
2
3
4
LAN
D
E
F
5
6
7
Bridge that is part of the spanning tree
H
J
8
9
Bridge that is not part of the spanning tree

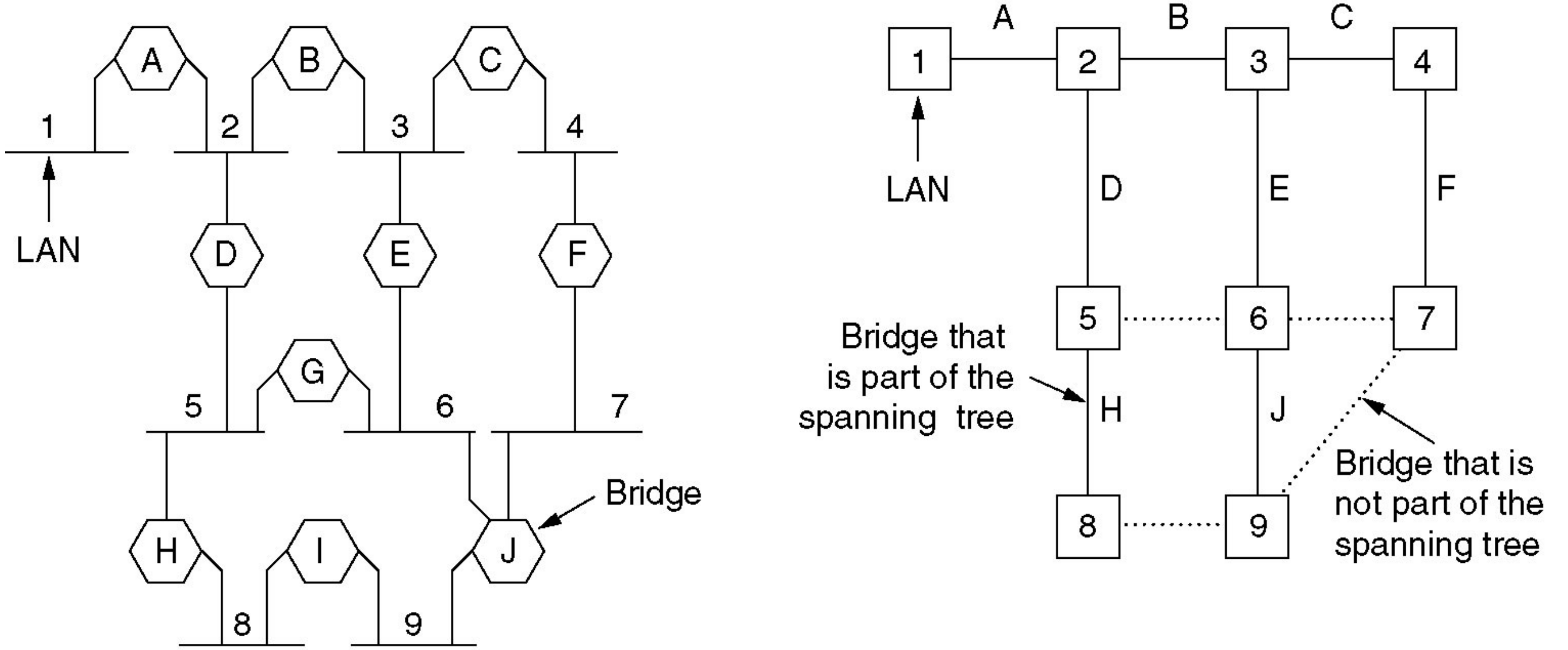
(Graphs are dual to each other)

Spanning tree algorithm from bridges

- Invented by Radia Perlman, IEEE 802.1D

- Bridges elect a leader (root)
  - Broadcast serial numbers, pick lowest
- Construct tree based at the root
  - Follow links out from root
- Packets forwarded along resulting spanning tree
- Continuous tree maintenance (and leader election)
- Allows redundant paths for fault tolerance

Redundancy


A
B
C
1
2
3
4
LAN
D
E
F
G
5
6
7
Bridge
H
I
J
8
9
A
B
C
1
2
3
4
LAN
D
E
F
5
6
7
Bridge that is part of the spanning tree
H
J
Bridge that is not part of the spanning tree
8
9

Redundant bridges/links

can be activated in the event of failure

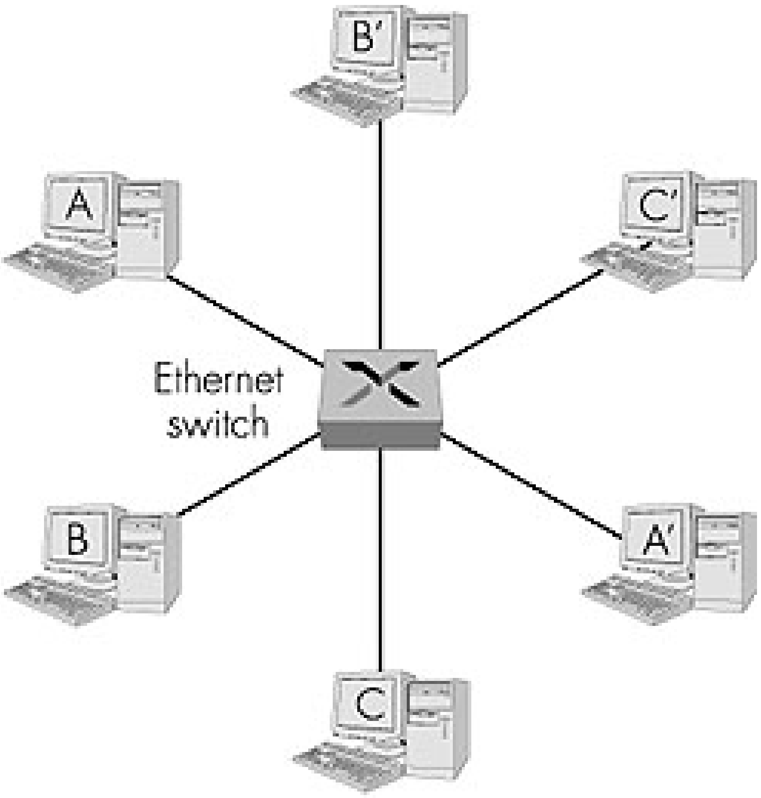
Some other bridge features

- Isolates collision domains resulting in higher total maximum throughput
- Limitless number of nodes and geographical coverage
- Can connect different Ethernet types
- Transparent (“plug-and-play”): no configuration necessary

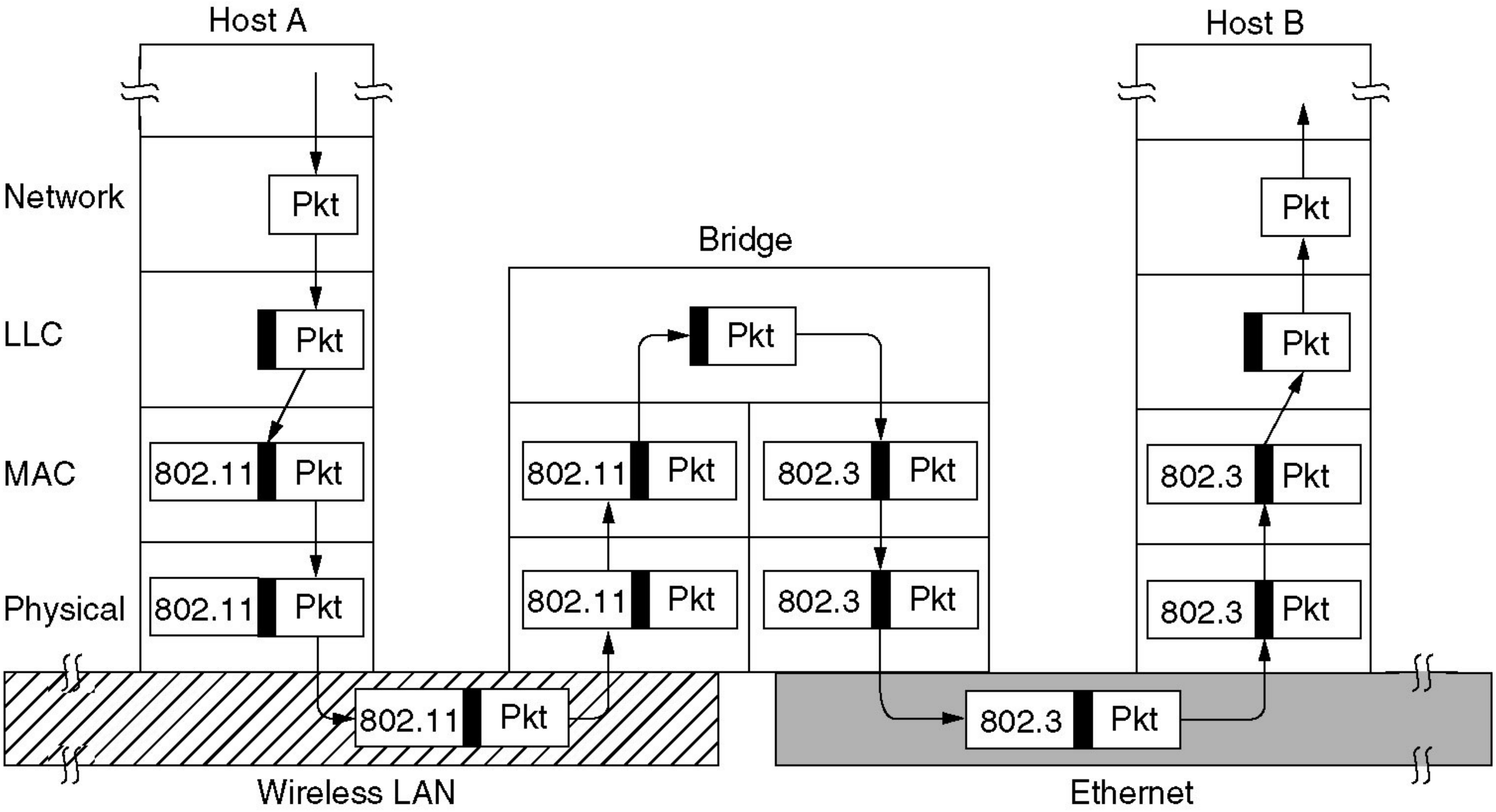
- Switches
  - Remark: switches are essentially multi-port bridges.
  - What we say about bridges also holds for switches!

Ethernet Switch

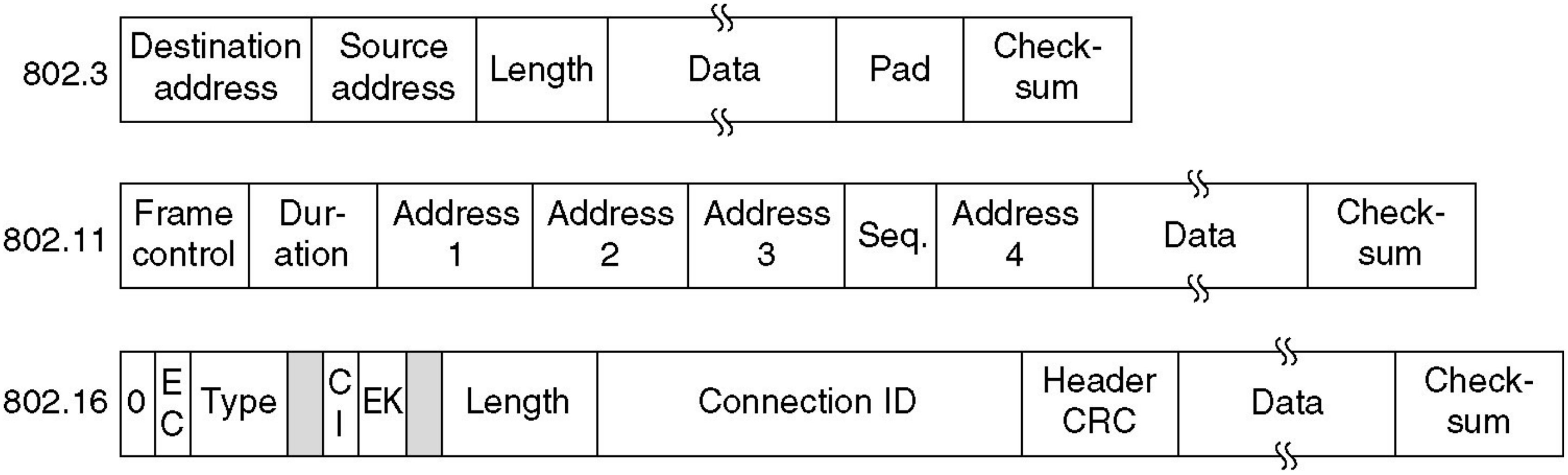
- Essentially a multi-interface bridge
- Layer 2 (frame) forwarding, filtering using LAN addresses
- Switching: A-to-A’ and B-to-B’ simultaneously, no collisions
- Large number of interfaces
- Often
  - Hosts star-connected into switch
  - Ethernet, but no collisions!
  - cut-through switching: forwarding without waiting for entire frame
  - combinations of shared/dedicated, 10/100/1000 Mbps interfaces



Multiprotocol bridges



Multiprotocol bridging: harder than it looks



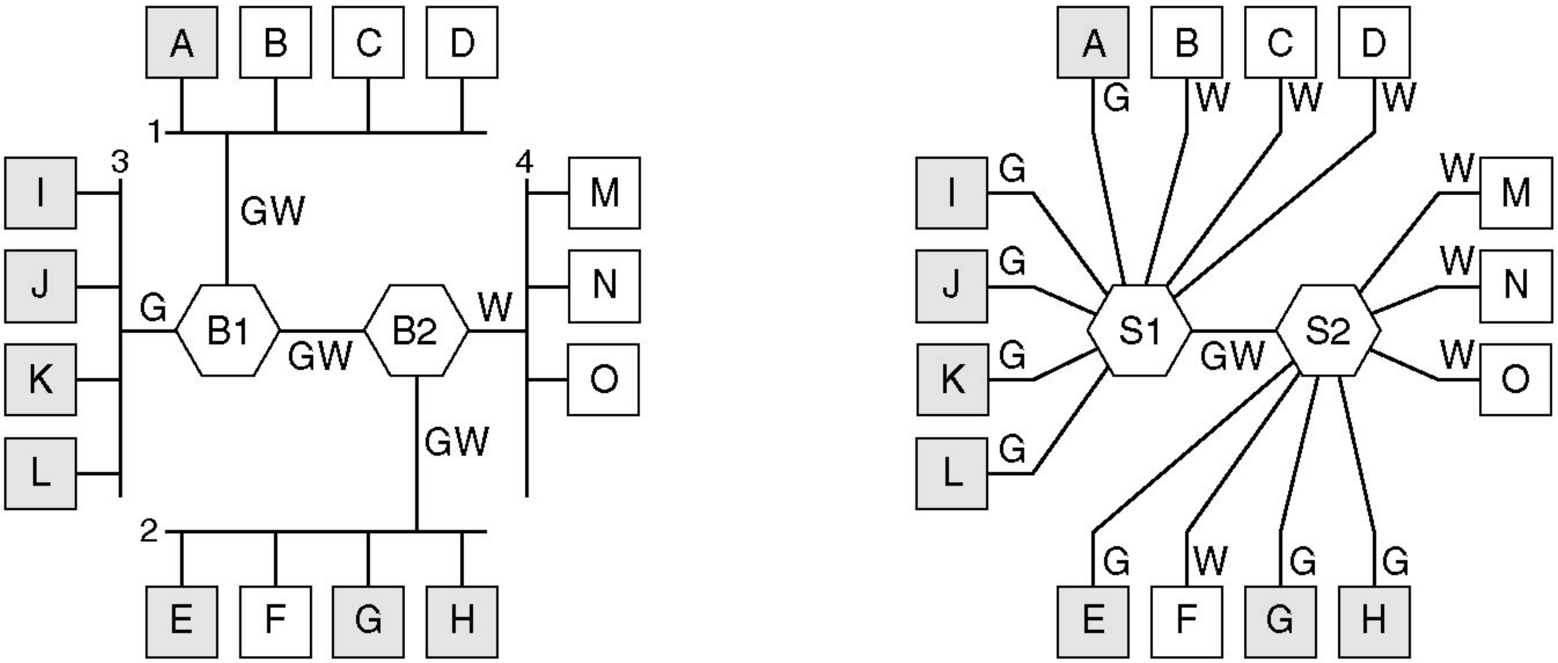
802.3: Destination address | Source address | Length | Data | Pad | Checksum

802.11: Frame control | Duration | Address 1 | Address 2 | Address 3 | Seq. | Address 4 | Data | Checksum

802.16: 0 | EC | Type | | CI | EK | | Length | Connection ID | Header CRC | Data | Checksum

- Different packet formats (even within IEEE 802)
- Link Encryption
- Link Retries
- Checksums

Virtual LANs (VLANS)



(a) (b)

- Key idea: make a set of switches look like a larger set of switches
- *Route* IP traffic between virtual LANs

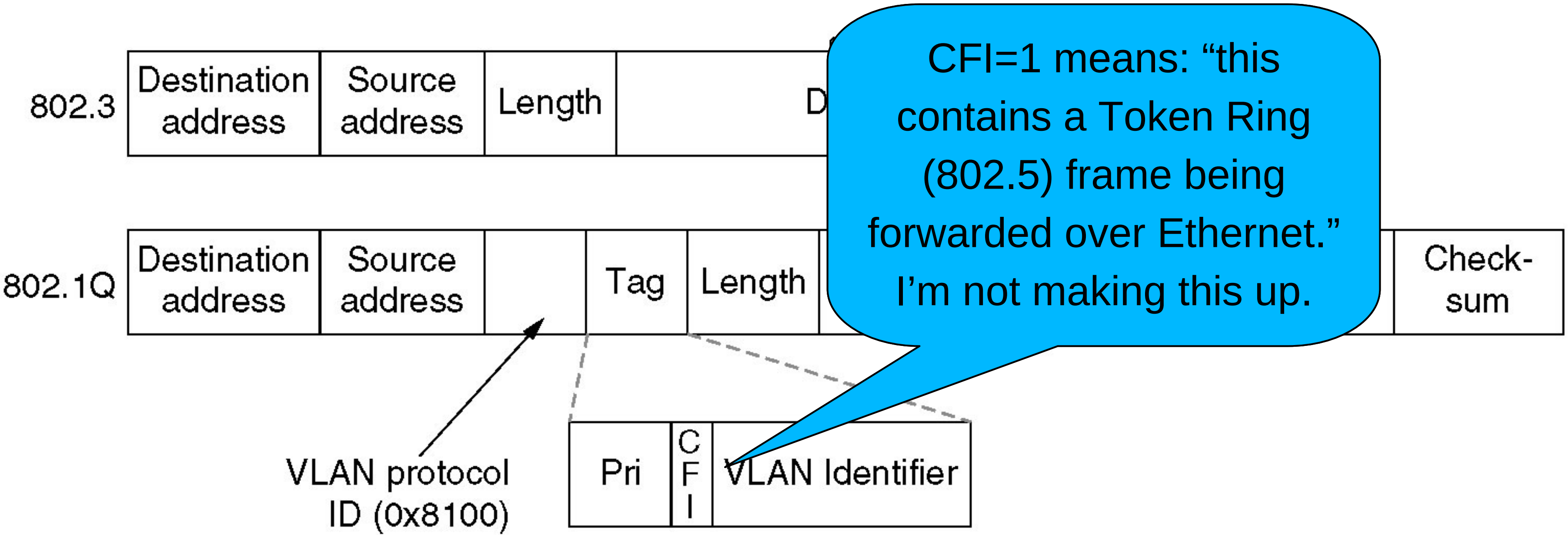
VLANs: Why and How?

- Why?
  - Security: isolate networks without separate wiring
  - Limit broadcasting (even switches forward all broadcasts)
  - Rapidly reconfigure network connections in software

- How?
  - Switch assigns each port to different VLAN
  - Switch assigns each MAC address to a VLAN
  - VLAN based on packet classification (e.g. protocol)
  - Explicit tag in each packet

- How to connect VLANs?
  - IP routing!

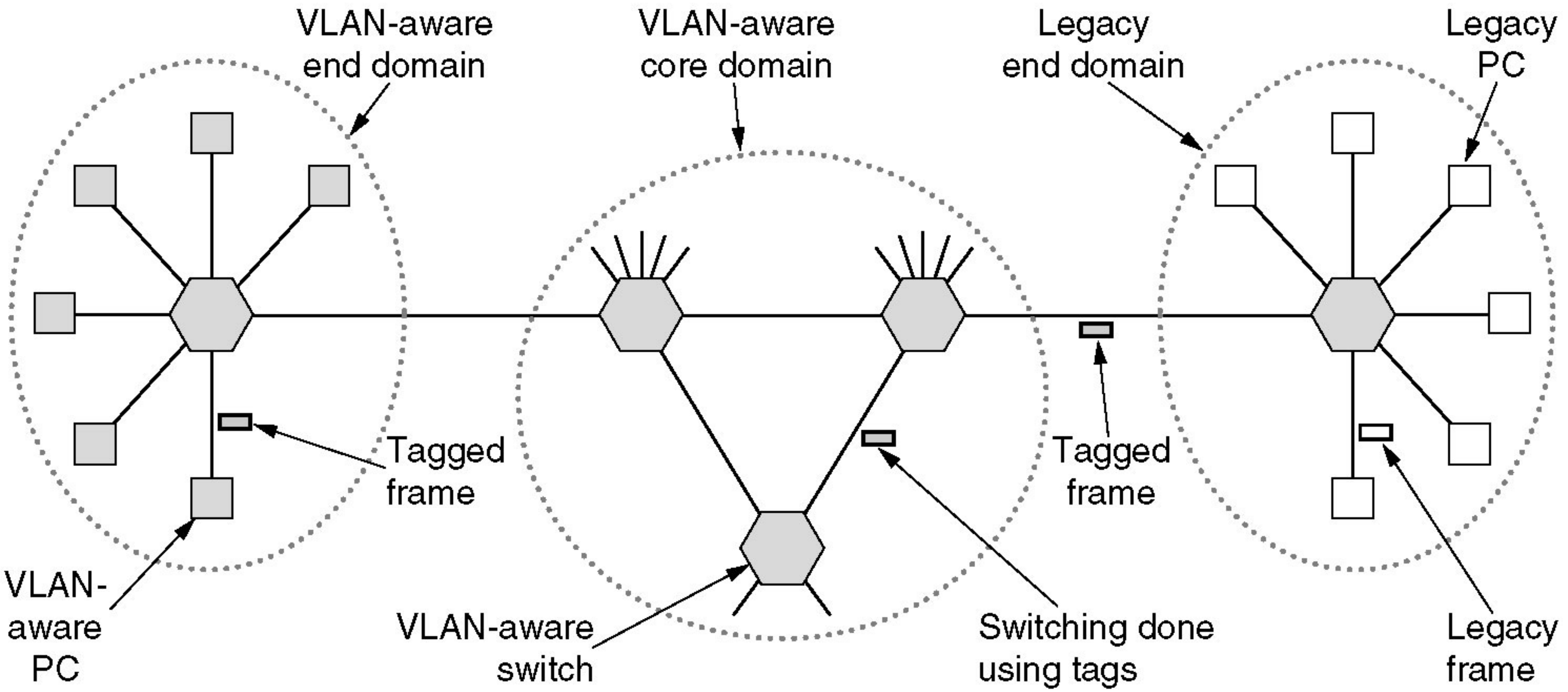
802.1Q (and 802.1p) Header



CFI=1 means: "this contains a Token Ring (802.5) frame being forwarded over Ethernet." I'm not making this up.

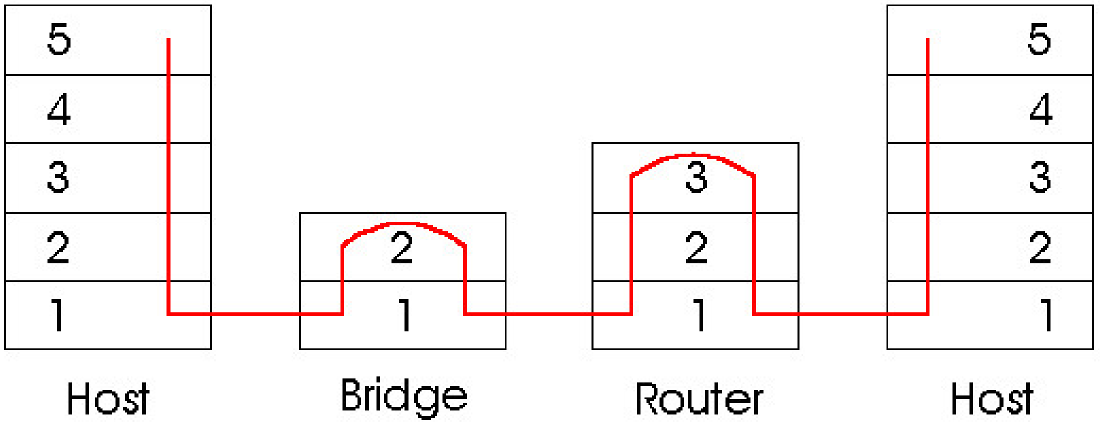
- How to tell 802.3 from 802.1Q?
  - Why is this unambiguous?

802.1Q VLAN example



Bridges vs. Routers

- both store-and-forward devices
  - routers: network layer devices (examine network layer headers)
  - bridges are link layer devices
- routers maintain routing tables, implement routing algorithms
- bridges maintain bridge tables, implement filtering, learning and spanning tree algorithms



Bridges vs. Routers

Bridges

- + Bridge operation is simpler requiring less packet processing
- + Bridge tables are self learning
- – All traffic confined to spanning tree, even when alternative bandwidth is available
- – Bridges do not offer protection from broadcast storms

Routers

- + arbitrary topologies can be supported, cycling is limited by TTL counters (and good routing protocols)
- + provide protection against broadcast storms
- - require IP address configuration (not plug and play)
- - require higher packet processing

bridges do well in small (few hundred hosts) while routers used in large networks (thousands of hosts)

Summary comparison

	hubs	bridges	routers	switches
traffic isolation	no	yes	yes	yes
plug & play	yes	yes	no	yes
optimal routing	no	no	yes	no
cut through	yes	no	no	yes

But…

Hardware reality

- All-in-one box: IP Routing Switch
  - Sometimes called a “Brouter” (!)
  - Ethernet, VLANs, IP, etc.
  - IP forwarding, multicast, etc.
  - Routing: RIP, OSPF, BGP,
  - Policy routing
  - Etc. etc.

- Question: where are the layers any more?

Point to Point Data Link Control

- one sender, one receiver, one link: easier than broadcast link
  - no Media Access Control
  - no need for explicit MAC addressing
  - e.g., dialup link, ISDN line
- popular point-to-point DLC protocols
  - PPP (point-to-point protocol)
  - HDLC: High level data link control (Data link used to be considered “high layer” in protocol stack!)

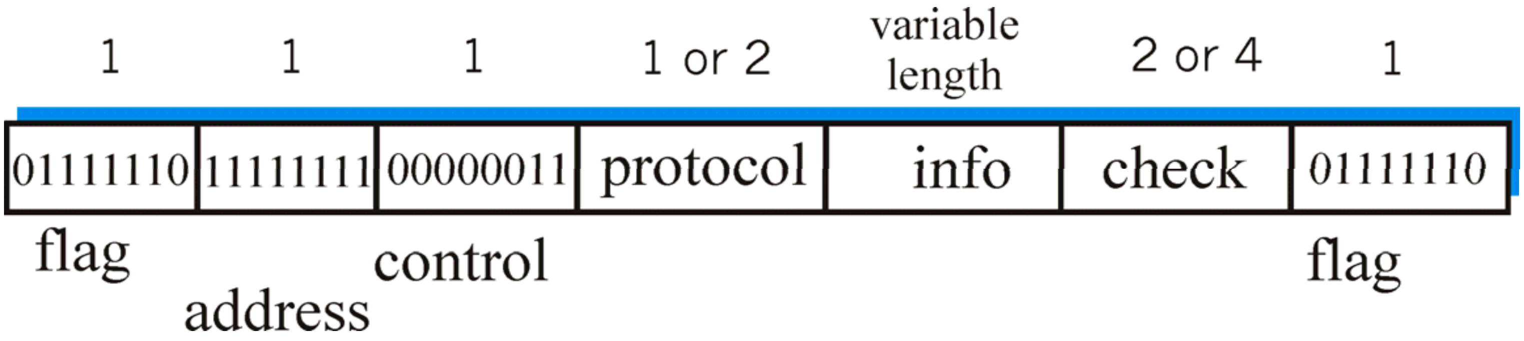
PPP Design Requirements [RFC 1557]

- packet framing
  - encapsulation of network-layer datagram in data link frame
  - carry network layer data of any network layer protocol (not just IP) *at same time*
  - ability to demultiplex upwards
- bit transparency: must carry any bit pattern in the data field
- error detection (no correction)
- connection liveness: detect, signal link failure to network layer
- network layer address negotiation: endpoints can learn/configure each other's network addresses

- No error correction/recovery, flow control, in-order delivery
  - all relegated to higher layers

PPP Data Frame

- Flag: delimiter (framing)
- Address: does nothing (only one option)
- Control: does nothing; in the future possible multiple control fields
- Protocol: upper layer protocol to which frame delivered
  (e.g. PPP-LCP, IP, IPCP, etc.)
- info: upper layer data being carried
- check: cyclic redundancy check for error detection



Byte Stuffing

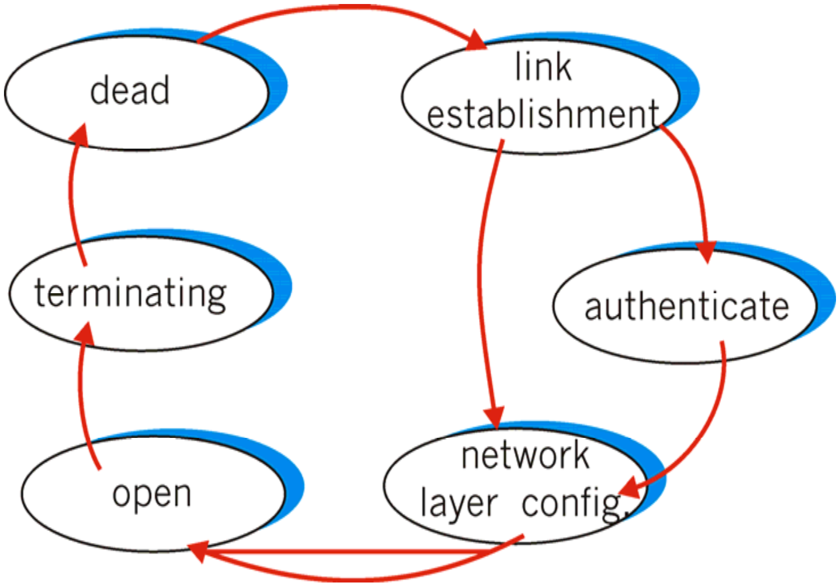
- “data transparency” requirement:
  - data must be allowed to include flag pattern <01111110>
  - Question: is received <01111110> data or flag?

- Sender: adds (“stuffs”) extra <01111110> byte after each <01111110> *data* byte
- Receiver:
  - two 01111110 bytes in a row: discard first byte, continue data reception
  - single 01111110: that’s the flag byte

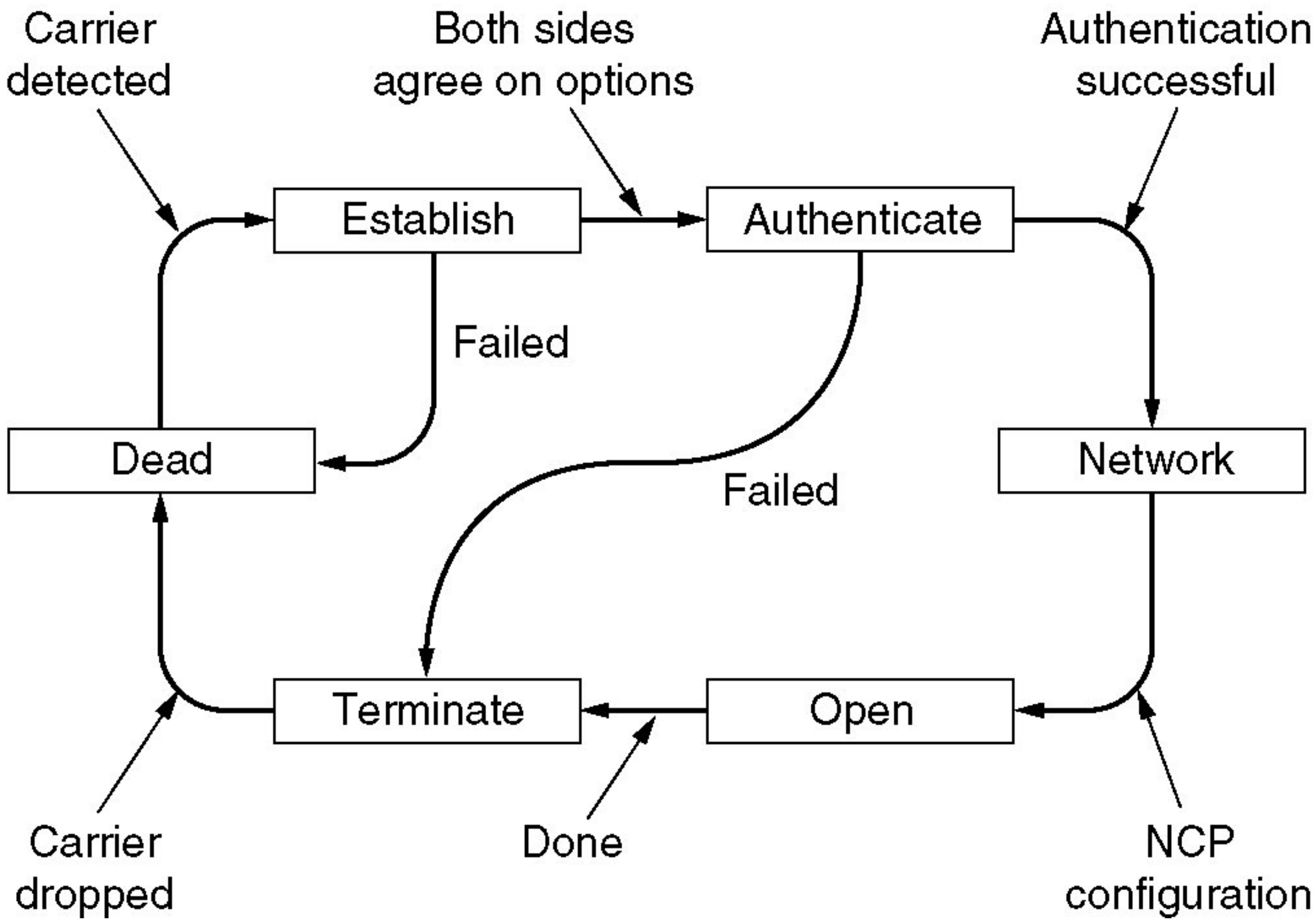
PPP Data Control Protocol

Before exchanging network-layer data, data link peers must

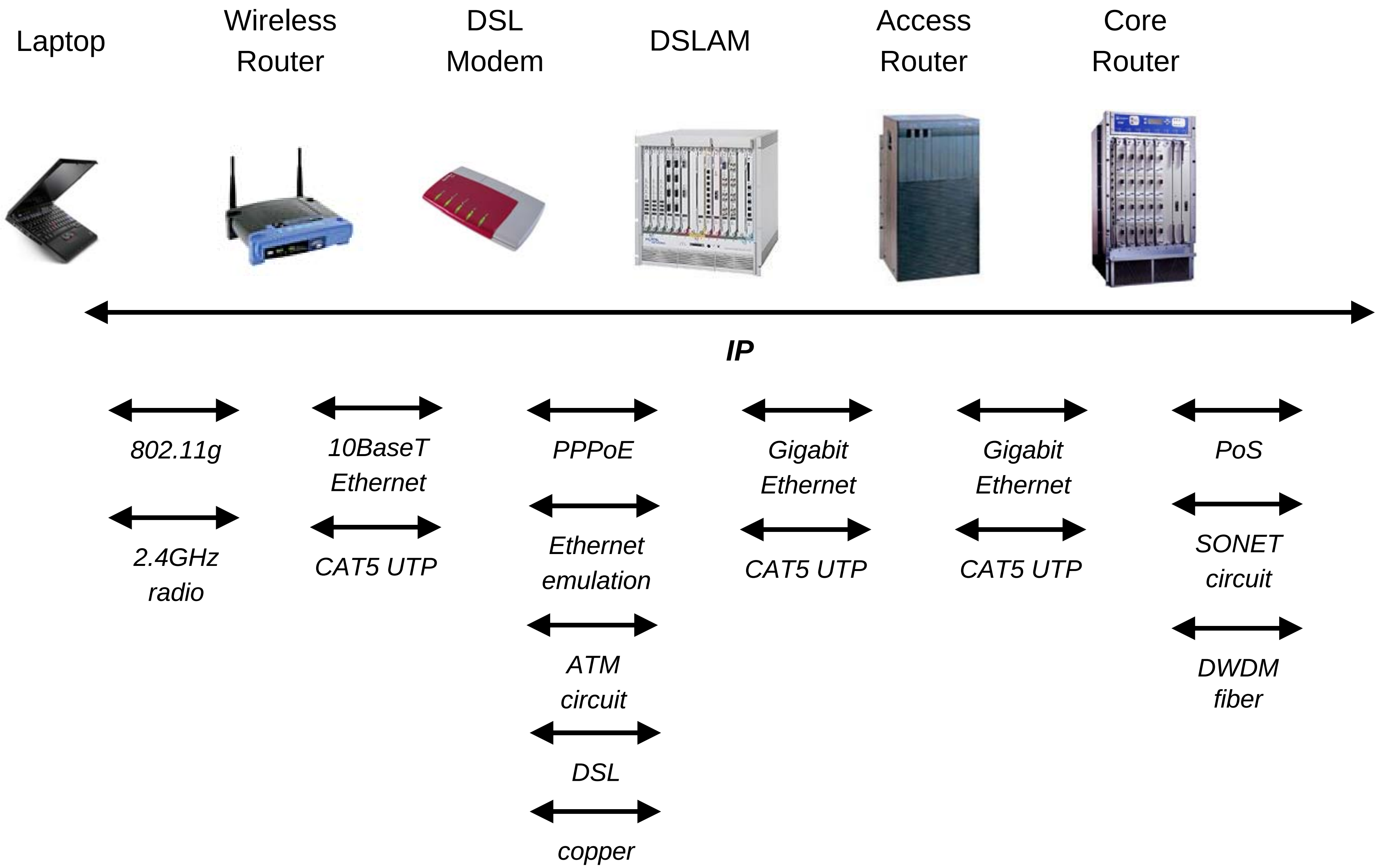
- configure PPP link
  - max. frame length
  - authentication
- learn/configure network layer information
  - for IP: carry IP Control Protocol (IPCP) msgs (protocol field: 8021) to configure/learn IP address



PPP state diagram



Half an IP end-to-end path to a web server



Laptop

Wireless Router

DSL Modem

DSLAM

Access Router

Core Router

IP

802.11g

2.4GHz radio

10BaseT Ethernet

CAT5 UTP

PPPoE

Ethernet emulation

ATM circuit

DSL

copper

Gigabit Ethernet

CAT5 UTP

Gigabit Ethernet

CAT5 UTP

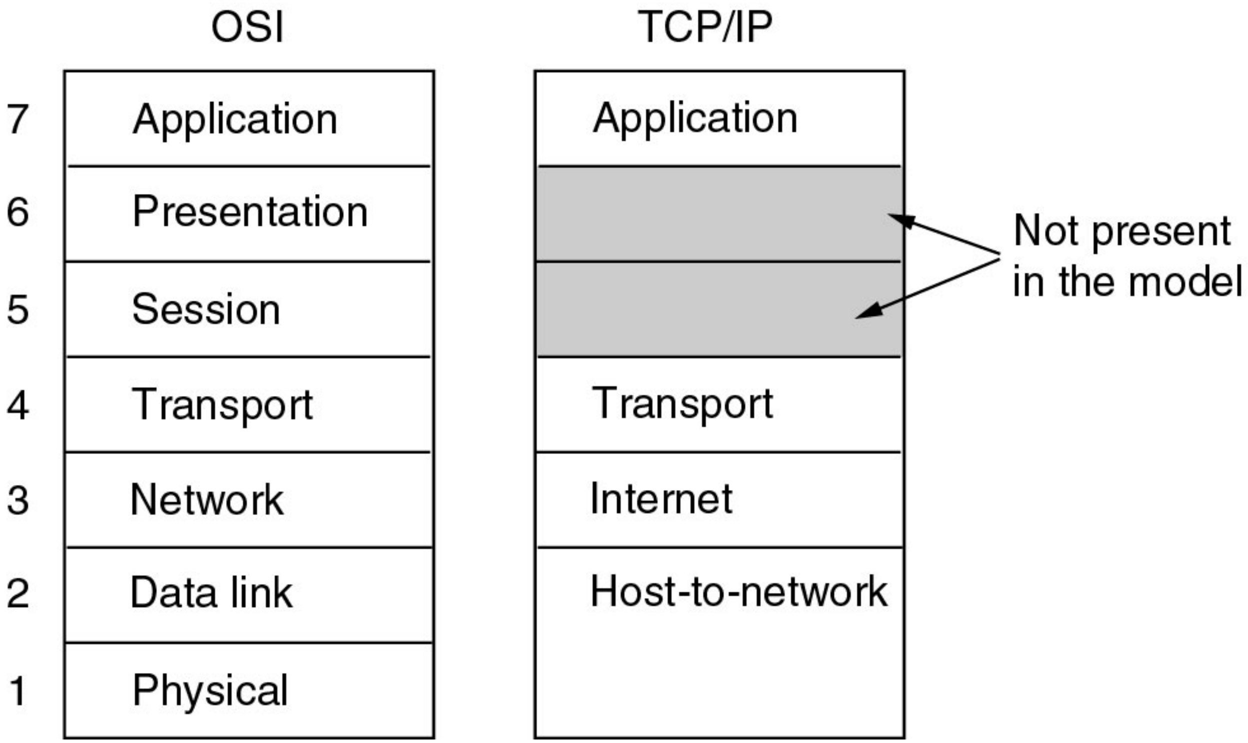
PoS

SONET circuit

DWDM fiber

Moral

- Recall our layering models:



	OSI	TCP/IP
7	Application	Application
6	Presentation	
5	Session	
4	Transport	Transport
3	Network	Internet
2	Data link	Host-to-network
1	Physical	

Not present in the model

Question: now we know what we know, are these useful?

Moral (contd):

- Layers *are* useful
  - – How else to talk about protocols?
  - – Separation of function is important
  - – Interoperability occurs at a layer
    - Translation at the layer below!
- Layers include encapsulation
  - ⇒ New layers can be inserted
  - ⇒ Layers can be “looped” (tunnelling) at any level
- Encapsulation can be broken
  - – “Deep packet inspection”, combined routing/switching
  - – “Cross layer visibility” (expose underlying information (both fashionable research topics!)

Next week and beyond:

- Wireless protocols
  - WiFi, WiMax, Bluetooth, etc.
- Introduction to distributed algorithms
  - Consensus, etc.
- Overlays and P2P networks
  - Yet more layer violations ☺
- More networking reality
  - ISP operations
  - Network management, Traffic engineering
- “Cool stuff”
  - Current hot research topics
  - Future directions in networking